Mobile CSP | Student Lesson

Unit 4 | ***Coin Flip Experiment***

Course Listing: http://course.mobilecsp.org

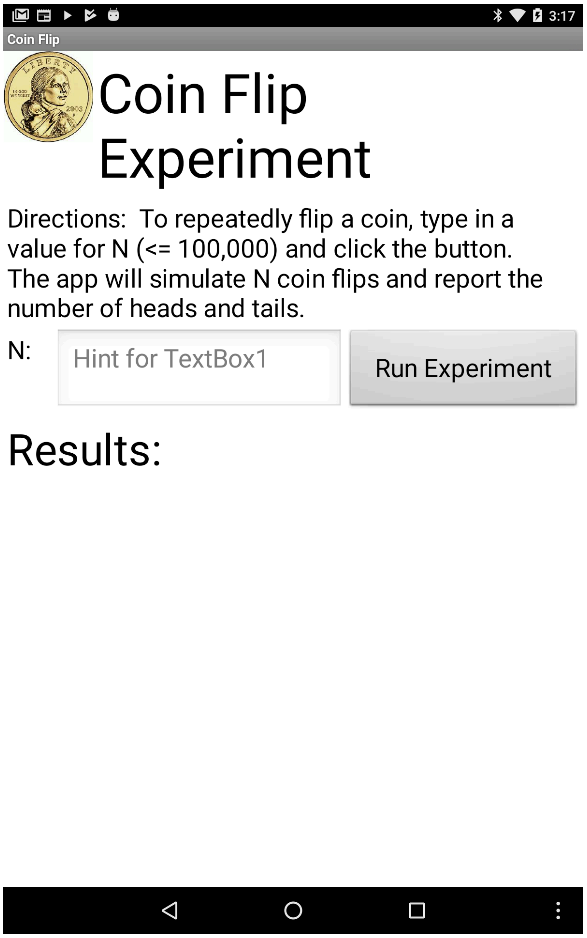
In this lesson you will use a slightly modified version of the app that you built in the preceding lesson, the *Coin Flip Simulation Tutorial.*

The *CoinFlipExperiment* app will let you conduct an experiment aimed at determining how 'good' App Inventor's PRNG is. The app lets the user 'flip a coin' *N* times and displays the results.

You should record and tally the results and calculate the average percentage of heads. The expectation is that as *N* gets large, the average should approach 50%.

Objectives: In this lesson you will learn to :

- use simulation software to conduct an experiment;
- make and test a hypothesis about App Inventor's ability to generate random numbers.



Coin Flip Experiment

Directions: To repeatedly flip a coin, type in a value for N (<= 100,000) and click the button. The app will simulate N coin flips and report the number of heads and tails.

Click to watch Preview Video

Getting Ready

There is no coding required for this lesson. But we will be using an app to help us explore how good App Inventor's random number functions are. You will be installing the app on your device. See below.

The Coin Flip Experiment App

The *CoinFlipExperiment* app is a modification of the app you built in the preceding lesson, the *Coin Flip Simulation Tutorial.* It differs is the following ways:

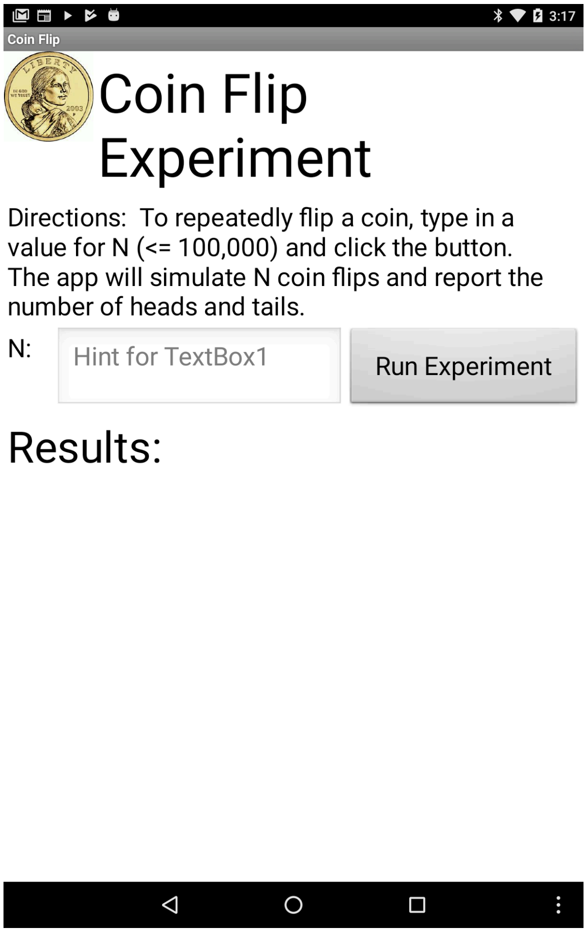
1. It incorporates some error-checking code to make sure the app doesn't crash if the user fails to type a number into the input textbox. And it guards against too large a number

being input. The limit is N <= 100,000.

2. It provides some 'Directions' for the user.

If you want to look at the app's source code, open App Inventor with the Coin Flip Experiment app.

The Experiment

 Coin Flip Coin Flip Experiment Directions: To repeatedly flip a coin, type in a value for N (<= 100,000) and click the button. The app will simulate N coin flips and report the number of heads and tails. N: Hint for TextBox1 Results:	As you can see from the screenshot, the app lets you specify the number of coin flips, *N.* It then flips the coin *N* times and reports the results as number of heads and number of tails. The experiment can be done individually or in pairs or the class as a group.

Mobile CSP | Student Lesson

Unit 4 | ***Coin Flip Experiment***

Course Listing: http://course.mobilecsp.org

Install the app

Scan the barcode below to download and install the app on your device.

Introduction

Here are some things you should know about how computers and computer languages (App Inventor) implement randomness:

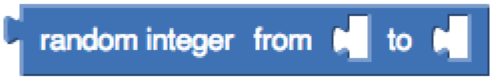
- A ***model*** is an abstract representation of some real world thing or phenomenon. An example would be our *Coin flip app*, which modeled the act of flipping a coin. Our model involves *randomness* in the form of App Inventor's ***random integer block.*** But that too is a model -- an abstract representation of a random process.
- Real examples of random processes would be processes such as flipping a real coin, or rolling a pair of dice, or drawing a card from a shuffled deck, or picking a ping pong ball out of an air chamber (like they do when they pick the lottery numbers).
- Randomness is used in lots of programs, especially games (Android Mash) and simulations (Coin Flip, 4-bit Computer Simulator).
- Creating a truly random process is hard to do. And creating a truly random process in a computer is no exception.
- Because creating true randomness is hard, computers use algorithms known as **pseudo random number generators (PRNGs)** to simulate randomness. This is much easier to do than generating truly random numbers. If you are curious about how PRNGs work, the next lesson goes into the details.
- PRNGs generate a sequence of "random seeming" numbers.
- PRNGs are models of true randomness. As such, they can be 'good' or 'bad' depending

on how well they approximate true randomness. Much research by mathematicians and computer scientists goes into creating good PRNGs.

- App Inventor uses a standard and well established PRNG, which should do a good job of modeling randomness.
- In this lesson we want to try to ***validate*** App Inventor's model for randomness -- i.e., we want to validate how well the random integer block ***approximates*** a truly random process.

The Experiment

Our CoinFlip app simulates flipping a coin. If you had a fair coin and you flipped it many, many times -- maybe a million times -- then if it were truly fair, you would expect it to come up "Heads" half the time. That's why we say for any coin flip, it has a 50:50 chance of coming up heads. App Inventor's random integer block uses its PRNG to generate a random sequence of integers. In our app, the sequence is between 1 and 2 inclusive. So, if the PRNG is good, it should generate a 1 half the time and a 2 half the time. And this, in turn, should let our Coin Flip app be a good model of flipping a coin.



random integer from to

Hypothesis

Our ***hypothesis*** is that App Inventor's *random integer block* is a good approximation of the process of randomly generating a 1 half the time and a 2 half the time.

If you were testing that a particular coin was "fair", you would flip it lots of times and record the number of heads and tails. Their ratio should come out 50:50. But you have to do a lot of flips.

So, to test our hypothesis about App Inventor's random integer block, we have to perform a simulated "coin flip" lots of times. To help with this, we will use the *Coin Flip Experiment* app, which will let us repeatedly "flip" a coin. The app uses an *algorithm* that uses the *random integer block.* If the random integer block is a good approximation of randomness, we would expect that when it is used to model the process of flipping a coin, it would make the odds of getting a "Heads" or "Tails" 50:50.

For our hypothesis to be true, the ratio between "Heads" and "Tails" in the app should approximate 50:50 as the number of coin flips gets large. The more coin flips we perform, the closer our ratio should be to 50:50.

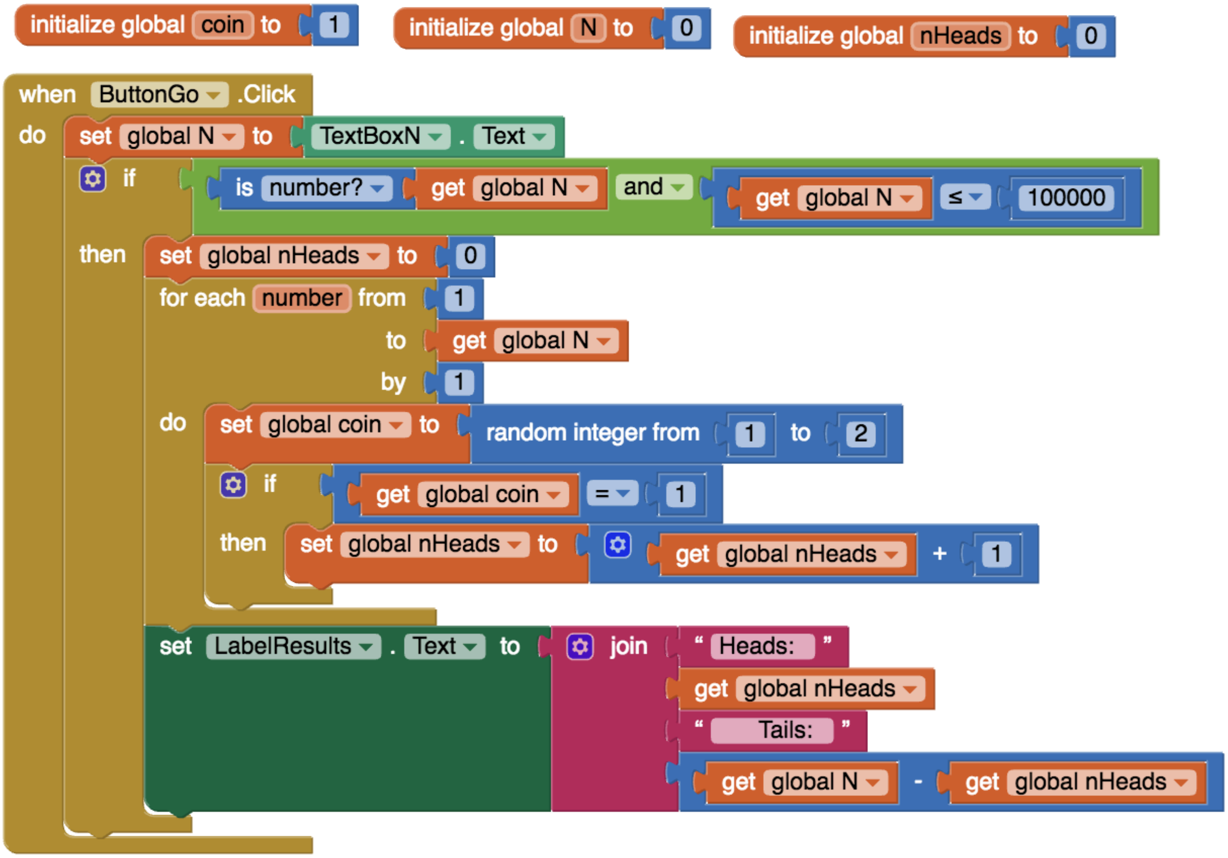
If the ratio does approach 50:50, that would ***validate*** our hypothesis. If it does not, that would prove that our hypothesis is ***invalid.***

Experimental Procedure

1. Repeatedly run the app and record the number of heads and tails received in each run. Do at least 10 runs. The app limits each run to <= 100,000 coin flips. 100,000 coin flips will likely take more than 1 minute to complete -- so you'll have to be patient.
2. Tally your results and calculate the percentage of heads for 10,000 coin flips, 20,000 coin flips, 100,000 coin flips or more.
3. Create a page named *Coin Flip Experiment* in your portfolio and record your results. Use a table to keep your data well organized. When you are done recording your results, complete the reflection questions provided below.

Reading the Source Code

Here is the source code for the app that is performing the experiment. As you can see, it is only slightly different from the version you created in the tutorial. The difference is an ***if*** statement after inputting *N* from the text box. The statement checks that *N* is a number (not the empty string) and that it's no greater than 100,000.



```
initialize global coin to 1
initialize global N to 0
initialize global nHeads to 0

when ButtonGo .Click
do  set global N to TextBoxN . Text
    if    is number? get global N  and  get global N ≤ 100000
    then  set global nHeads to 0
          for each number from 1
                            to get global N
                            by 1
          do  set global coin to random integer from 1 to 2
              if    get global coin = 1
              then  set global nHeads to get global nHeads + 1
          set LabelResults . Text to join " Heads: "
                                          get global nHeads
                                          " Tails: "
                                          get global N - get global nHeads
```

Nice work! Complete the Self-Check Exercises and Portfolio Reflection Questions as directed by your instructor.